

Projet individuel asp.net C#

Annuaire entreprise

Table des matières

Introduction.....	3
Présentation de l'équipe	3
Présentation du projet	3
Conception du projet	4
Prémices	4
Modélisation.....	5
Réalisation du projet	10
Langage de programmation	10
Librairies	10
Outils	11
Structure du projet.....	11
Présentation du projet	12
Les principales fonctionnalités	12
Bilan.....	15
Difficultés rencontrées	15
Axes d'améliorations	15
Conclusion	16

Introduction

Présentation de l'équipe

L'équipe est composée d'un développeur missionné pour répondre à un besoin de réalisation d'un site web de type annuaire d'entreprise, le développeur est moi même : Nicolas Vanhuysse.

Présentation du projet

Le sujet

Je suis missionné pour concevoir un site web de type annuaire entreprise.

Le sujet qui m'a été confié se formule ainsi :

Vous êtes développeur dans une industrie agro-alimentaire qui s'étend sur le territoire Français.

L'entreprise est en croissance et aujourd'hui elle est composée de

- 5 sites dans différentes villes :
 - Paris : Siège administratif
 - Nantes : site de Production
 - Toulouse : site de Production
 - Nice : site de Production
 - Lille : site de Production
- Plus de 1000 salariés.

A ce titre il est nécessaire de fournir aux utilisateur une application permettant de connaître instantanément le numéro de téléphone d'un collaborateur.

Les rôles dans l'application seront :

- Visiteur : Afficher les informations, recherche
- Administration : Ajout / Modification / Suppression d'information. Un accès « secret » à l'application via une combinaison de touche qui demande ensuite un mot de passe.

L'administrateur doit pouvoir faire :

- CRUD des sites : champ « ville » uniquement
- CRUD des services : Comptabilité, production, accueil, informatique, commercial etc.
- CRUD des salariés : Nom, prénom, téléphone fixe, téléphone portable, email, service (association), site (association)

Le visiteur doit pouvoir afficher la fiche salariée et réaliser une recherche :

- Par nom de salarié : saisir X lettres pour afficher les salariés correspondants
- Par site via une liste des sites
- Par service via une liste des services

La fiche salarié affiche toutes les informations du salarié (nom, prénom, téléphone fixe, téléphone portable, mail, service, site)

Objectifs

Les objectifs du projet qui m'est confié sont les suivants :

- Réaliser la conception d'un site web en accord avec le sujet : Un site web de type « Annuaire d'entreprise ».
- Pouvoir rechercher, coder débbugger et commenter son code
- Développer un site web permettant d'interagir avec les données d'une base de données (Select, Insert, Update, Delete).
- Configurer un serveur web en local ou à distance

Contraintes

Certaines contraintes ont été imposées pour réaliser ce projet, la liste est la suivante :

- Utiliser le langage : asp.Net
- Conception d'une base de données (MCD, MLD et script)
- Mise en place d'un ou plusieurs CRUD (Create, Read, Update, Delete)
- Mise en place d'un system de recherche selon 3 critères : Par site, par service et par nom.
- Gestion de l'accès en administrateur avec un contrôle d'accès.
- Le code est en Objet, il est bien organisé, suffisamment commenté.
- Un cahier de test complet doit être livré avec le projet.
- Code versionné sur Git.

Conception du projet

Prémices

J'ai commencé par réfléchir au sujet et pris des décisions stratégiques pour le projet. La première étape : trouver un nom.

Annuaire

Le choix du nom du projet a été décidé : Annuaire. Le nom est évocateur et correspond bien au site web que je souhaite développer.

Github :

Une fois le nom trouvé, il est temps de paramétrer un repository github afin versionner mon travail, le lien de mon projet est donc le suivant :

<https://github.com/nicolasvanhuyse/ProjetAnnuaire>

Modélisation

Dictionnaire de données

Avant de commencer la modélisation de la base de données, il est très important pour moi de mettre en place un dictionnaire de données qui transcrira à l'écrit l'ensemble des actions qu'il sera possible d'effectuer sur le site web. En regroupant les mots clés qui résulteront de cet exercice il ne nous restera plus qu'à les retranscrire sous forme d'entités et de relations avec la méthode Merise.

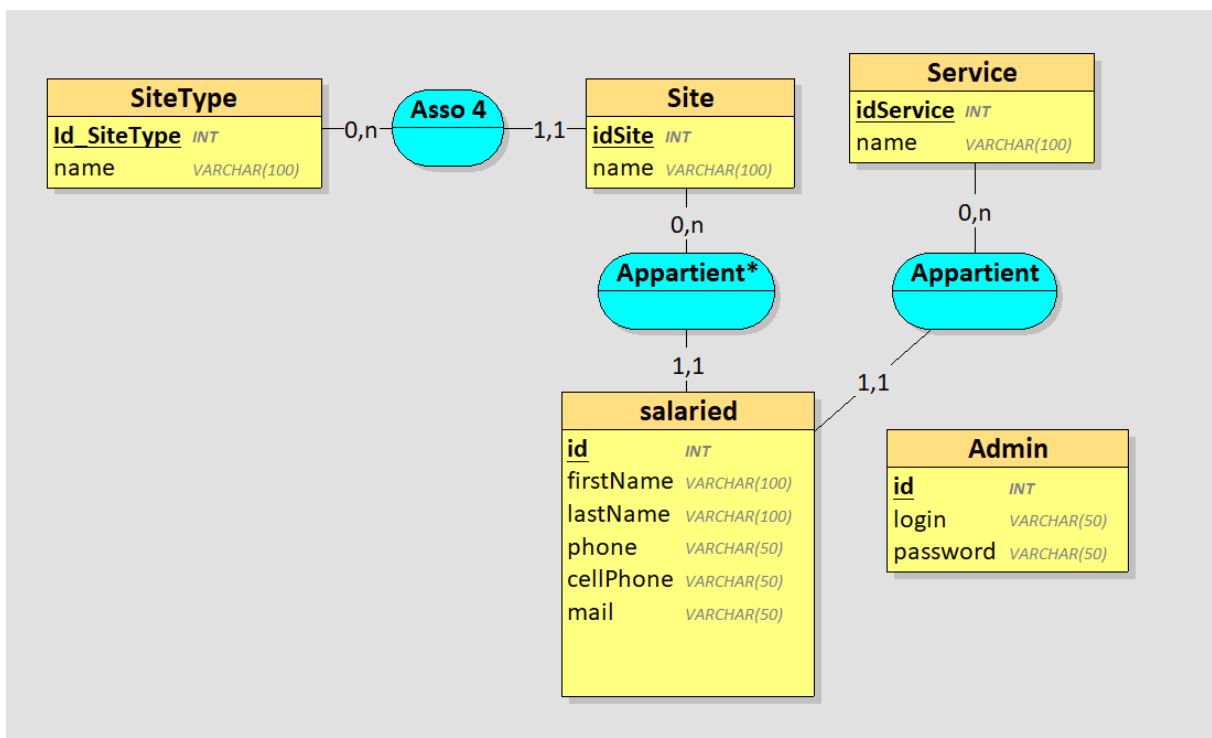
La formulation de notre dictionnaire de données est la suivante :

- Le visiteur peut afficher la fiche salariée et réaliser une recherche par nom, par site et par service
- L'Administrateur peut gérer les salariés, les sites et les services
- Les fiches des salariés possèdent : le nom du salarié, son prénom, son téléphone fixe, son téléphone portable, son mail, son service et son site.

MCD

Une fois le dictionnaire de données établi il est temps de se mettre à la modélisation de la base de données, pour cela j'ai utilisé le logiciel gratuit et libre d'utilisation [Looping](#) qui ne me limite pas en termes de création d'entité comme son concurrent JMerise.

Le MCD (Modèle Conceptuel de Données) créé est le suivant :



Modèle conceptuel de données du site Annuaire

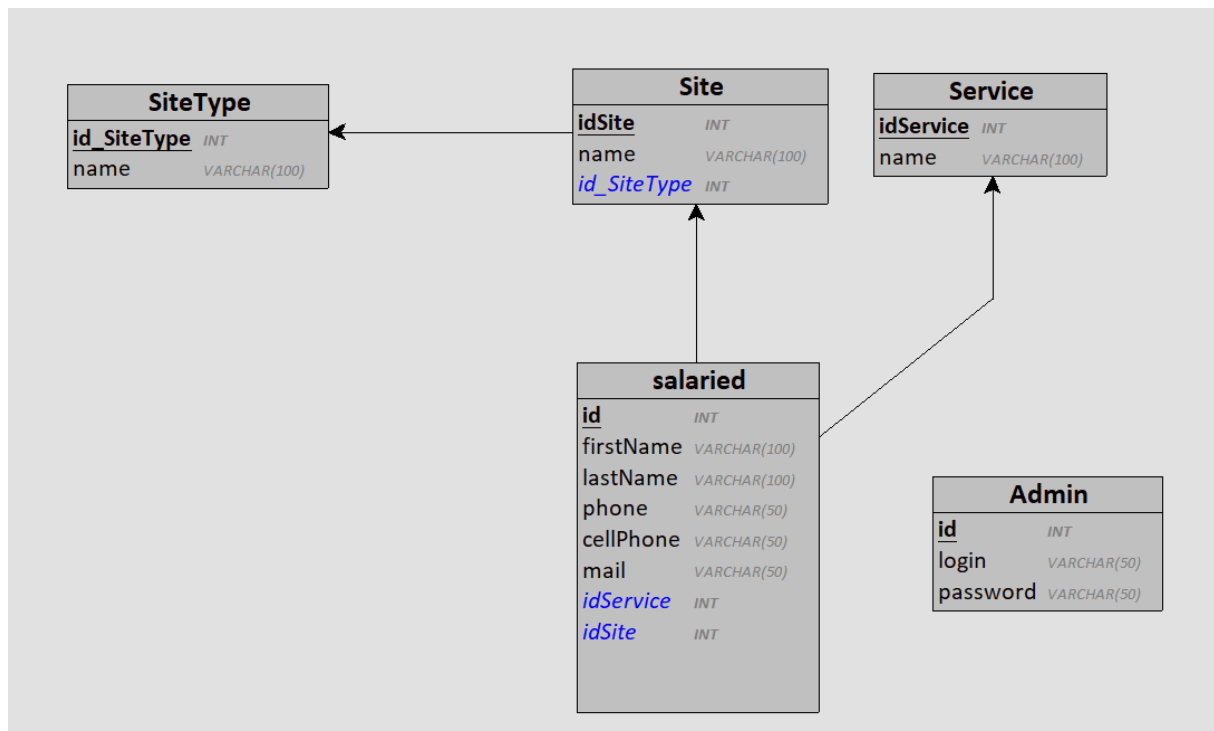
On retrouve bien l'ensemble des éléments cités dans le Dictionnaire de données et les associations qui sont liées aux entités.

Nous pouvons voir les différentes tables ainsi que les relations entre ces dernières et leurs cardinalités. Les cardinalités sont des liens entre une table et une relation précisant le minimum et le maximum qu'un objet de la table peut être concerné par la relation.

Il faut maintenant passer à la génération du MLD pour avoir une idée plus claire de ce à quoi va ressembler notre base de données finale.

MLD

L'avantage du logiciel Looping est qu'il permet d'un simple clic de passer de notre MCD à notre MLD (Modèle Logique de Données). Voici ce que l'on obtient :



Modèle logique de données du site Annuaire

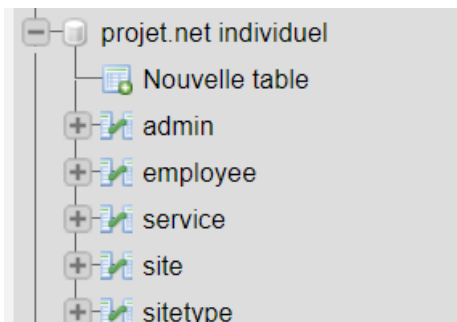
Sur ce modèle nous pouvons voir que les relations ont disparu et que selon les cardinalités définies dans le MCD, des « clés étrangères » sont apparues.

La structure obtenue nous permet déjà de mieux nous rendre compte de ce à quoi va ressembler notre base de données finale. Il ne nous reste plus qu'à créer le code SQL pour mettre en place tout cela, et encore une fois Looping nous permet de générer le script SQL en un clic en choisissant le type de SGBD que nous utilisons, dans notre cas c'est une base de données MySQL.

```
CREATE TABLE Service(  
    idService INT Auto_increment NOT NULL,  
    name VARCHAR(100),  
    PRIMARY KEY(idService)  
);  
  
CREATE TABLE Admin(  
    id INT Auto_increment NOT NULL,  
    login VARCHAR(50),  
    password VARCHAR(50),  
    PRIMARY KEY(id)  
);  
  
CREATE TABLE SiteType(  
    id_SiteType INT Auto_increment NOT NULL,  
    name VARCHAR(100),  
    PRIMARY KEY(id_SiteType)  
);  
  
CREATE TABLE Site(  
    idSite INT Auto_increment NOT NULL,  
    name VARCHAR(100),  
    id_SiteType INT NOT NULL,  
    PRIMARY KEY(idSite),  
    FOREIGN KEY(id_SiteType) REFERENCES SiteType(id_SiteType)  
);  
  
CREATE TABLE salaried(  
    id INT Auto_increment NOT NULL,  
    firstName VARCHAR(100),  
    lastName VARCHAR(100),  
    phone VARCHAR(50),  
    cellPhone VARCHAR(50),  
    mail VARCHAR(50),  
    idService INT NOT NULL,  
    idSite INT NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(idService) REFERENCES Service(idService),  
    FOREIGN KEY(idSite) REFERENCES Site(idSite)  
);
```

Script SQL généré par looping pour notre site web

Le script SQL est maintenant généré, il ne nous reste plus qu'à le lire dans notre application de gestion de base de données MySQL. L'application utilisée est PhpMyAdmin.



UML : Diagramme de cas d'utilisation

UML : « Unified Modeling Language » est le langage standard pour visualiser, spécifier, construire, et documenter des produits d'un système dont le logiciel est prépondérant.

J'ai commencé par faire un diagramme de cas d'utilisation, il représente les fonctionnalités (cas d'utilisation) nécessaires aux utilisateurs.

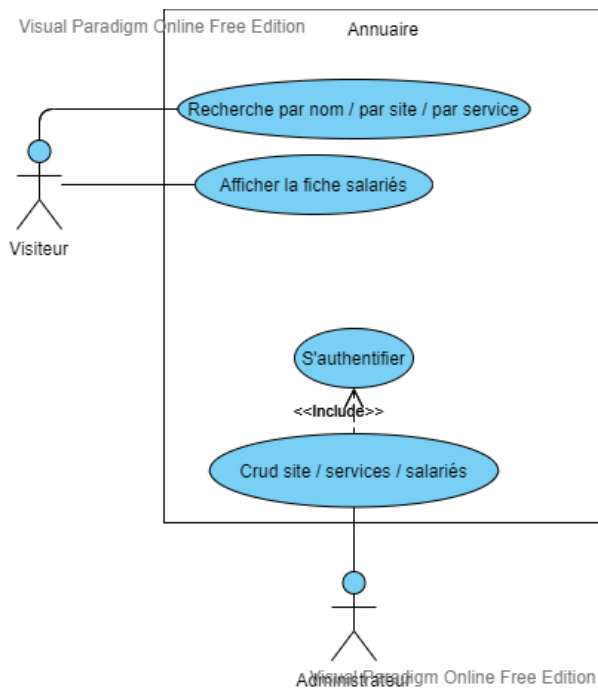


Diagramme de cas d'utilisation du site Annuaire

Sur ce diagramme de cas d'utilisation nous pouvons voir qu'un visiteur aura la possibilité d'effectuer des recherches par nom, par site et par service et qu'il pourra afficher la fiche d'un salarié.

L'administrateur quant à lui aura la possibilité de gérer les CRUD salarié, site et service mais il y aura forcément une authentification requise pour cela.

UML : Diagramme de classe

Le diagramme de classes montre la structure interne. Ce diagramme contient principalement des classes. Une classe contient des attributs et des opérations. Le diagramme n'indique pas comment

utiliser les opérations : c'est une description purement statique d'un système.

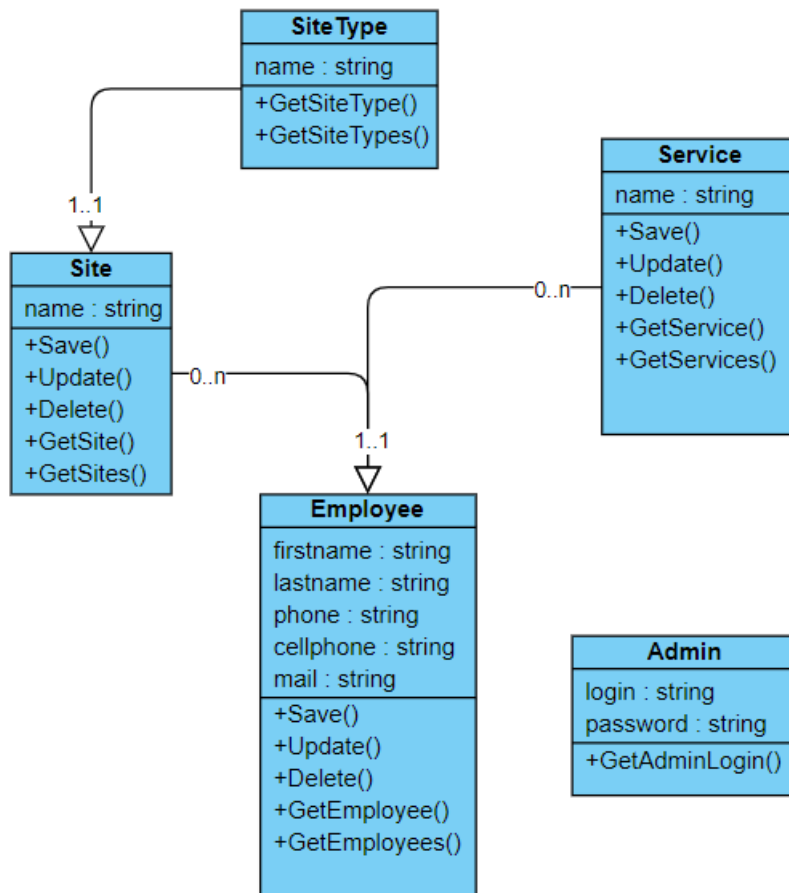


Diagramme de classe du site Annuaire

Sur ce diagramme nous pouvons voir les différentes classes du projet ainsi que leurs attributs et leurs opérations.

Par exemple : pour la classe « Site », nous pouvons voir qu'elle prend l'attribut « name » qui est un string et en opérations : Save(), Update(), Delete(), GetSite() et GetSites(). Ces opérations sont les fonctions qui seront utilisées par la classe « Site ».

Réalisation du projet

Pour réaliser le projet, j'ai utilisé ASP.NET qui est un Framework Web gratuit permettant de créer des sites Web et applications Web en utilisant HTML, CSS et JavaScript ainsi que C#.

Langage de programmation

Pour réaliser ce projet j'ai été amenés à utiliser un certain nombre de langages de programmation, je vais ici vous décrire chacun des langages utilisés et leur intérêt pour la réalisation du site web.

HTML – CSS

HTML « *HyperText Markup Language* » est un langage de balise permettant de représenter le contenu d'une page web ainsi que sa structure. Il est utilisé sur l'ensemble des sites web.

CSS « *Cascading Style Sheets* » est un langage permettant la mise en forme d'une page web. Il permet globalement d'appliquer des styles sur des éléments de balises HTML.

Ces deux langages ont été plus qu'essentiels dans la réalisation de notre site car il réalise le cœur de ce que verra l'utilisateur, à savoir le texte et le style de notre site.

C#

C# est un langage orienté objet qui utilise des syntaxes issues du C++ et très proches du Java. Il offre entre autres possibilités des techniques d'encapsulation, de polymorphisme et d'héritage de classe.

SQL

SQL est l'acronyme de *Structured Query Language*. Le SQL est un langage permettant de communiquer avec les bases de données relationnelles. Ce langage permet de manipuler les données et est généralement commun à tous les SGBD hormis certaines exceptions.

Librairies

Une librairie est un ensemble de fonction utilitaire, regroupée et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Dans ce projet, j'ai utilisé quelques librairies pour m'aider et rendre certaines tâches plus simples à réaliser. J'ai donc utilisé Bootstrap qui est un framework CSS et qui permet de rendre le site esthétique et adapté au format mobile (responsive) facilement et rapidement. J'ai également utilisé Icons Font Awesome qui permet d'utiliser différents icons pour des boutons par exemple.

Outils

Pour ce qui est des outils, J'ai utilisé Microsoft Visual Studio qui facilite le développement d'applications en Asp.net et qui me sert en même temps d'éditeur de code. Pour modéliser la base de données et sortir le MCD et MLD j'ai utilisé l'outil Looping. Pour réaliser les diagrammes UML, j'ai utilisé Visual Paradigm Online. Ensuite pour créer la base de données et pouvoir l'utiliser j'ai tout d'abord créé un serveur web en local grâce à XAMPP ce qui nous permet d'afficher notre site sur l'url <http://localhost/> et j'ai utilisé PhpMyAdmin qui est une application Web permettant la gestion d'une base de données utilisant le SQL. Pour ce qui est du versionning j'ai utilisé Git et plus particulièrement GitHub. Enfin, un cahier de test à été rédigé ce qui permet de tester toutes les fonctionnalités que l'on doit retrouver dans le projet et définir le résultat attendu.

Test Effectué	Résultat Attendu	Commentaires
CONNEXION		
1 / J'ouvre la page de connexion en ajoutant /admin à l'URL	La fenêtre connexion apparaît avec un champ login, mdp et un bouton connexion	OK
2 Le login et mdp sont valides et je clique sur connexion	Connexion acceptée avec une redirection sur la page admin avec un lien Admin et un bouton déconnexion ajouté dans la navbar	OK
3 Le login et mdp ne sont pas valides et je clique sur connexion	Connexion refusée redirection sur la même page avec un message : "Erreur de connexion"	OK
DECONNEXION		
1 Je suis connecté en admin et je clique sur le bouton de déconnexion	Déconnexion et redirection vers la page d'accueil	OK
ACCUEIL		
1 / J'ouvre la page d'accueil	Affichage de la liste des salariés avec la possibilité d'afficher le détail d'un salarié et de filtrer par nom, site ou service avec un bouton de recherche	OK
2 Filtre par nom et clique sur le bouton recherche	Affichage de la liste des salariés en fonction du filtre	OK
3 Filtre par site et clique sur le bouton recherche	Affichage de la liste des salariés en fonction du filtre	OK
4 Filtre par service et clique sur le bouton recherche	Affichage de la liste des salariés en fonction du filtre	OK
5 Filtre par nom, site et service et clique sur le bouton recherche	Affichage de la liste des salariés en fonction des filtres	OK
6 Clique sur le bouton pour afficher le détail d'un salarié	Redirection vers la page détail du salarié avec toutes les informations du salarié	OK
PAGE ADMIN		
1 Clique sur le lien gestion des salariés	Redirection vers la page gestion des salariés avec le tableau des salariés, un bouton ajouter et des boutons edit et supprimer	OK
2 Clique sur le lien gestion des services	Redirection vers la page gestion des services avec le tableau des services, un bouton ajouter et des boutons edit et supprimer	OK
3 Clique sur le lien gestion des sites	Redirection vers la page gestion des sites avec le tableau des sites, un bouton ajouter et des boutons edit et supprimer	OK
GESTION DES SALARIÉS		
1 / J'ouvre la page de gestion des salariés	Affichage d'un tableau avec la liste des salariés, un bouton ajouter et des boutons edit et supprimer sur chaque ligne du tableau	OK
2 Clique sur le bouton ajouter	Redirection vers la page d'ajout d'un salarié avec un formulaire avec tous les champs du salarié et un bouton enregistrer	OK
3 Clique sur le bouton edit	Redirection vers la page d'edit du salarié avec un formulaire rempli des informations du salarié et un bouton enregistrer	OK
4 Clique sur le bouton supprimer	Redirection vers la même page avec le salarié sélectionné supprimé	OK
AJOUT SALARIÉ		
1 / J'ouvre la page d'ajout d'un salarié	Affichage d'un formulaire avec tous les champs d'un salarié vide et un bouton enregistrer	OK
2 Les champs sont correctement remplis et je clique sur enregistrer	Redirection sur la page gestion des salariés avec le nouveau salarié créé	OK
3 Les champs ne sont pas correctement remplis et je clique sur enregistrer	Message d'erreur sur le champ non rempli	OK
EDIT SALARIÉ		
1 / J'ouvre la page d'edit du salarié	Affichage d'un formulaire avec tous les champs du salarié remplis et un bouton enregistrer	OK
2 Modification des champs et je clique sur enregistrer	Redirection vers la page gestion des salariés avec les nouvelles informations enregistrées	OK
3 Aucune modification et je clique sur enregistrer	Redirection vers la page gestion des salariés	OK
4 Champs vides et je clique sur enregistrer	Message d'erreur sur le champ non rempli	OK

Structure du projet

Le projet a été structuré en utilisant le MVC (Modèle, Vue, Contrôleur). Le MVC est une architecture logicielle populaire pour les applications web. Cela nous permet de bien organiser le code source de notre projet et définir le rôle de chaque dossier.

Pour le fonctionnement, la Vue appelle le Contrôleur qui exécute des fonctions se trouvant dans le Modèle.

master	ProjetAnnuaire / ProjetAnnuaire /	Go to file	Add file	...
nicolasvanhuyse	Ajout script sql + readme + Cahier de test	6e5ee5b	17 minutes ago	History
..				
.vs	Ajout script sql + readme + Cahier de test		17 minutes ago	
Controllers	control input		23 hours ago	
Models	Connexion admin		2 days ago	
Properties	update projet		last month	
Services	Debut session		last month	
ViewModels	update		last month	
Views	control input		23 hours ago	
bin/Debug/net5.0	Ajout script sql + readme + Cahier de test		17 minutes ago	
obj	Ajout script sql + readme + Cahier de test		17 minutes ago	
wwwroot	Init projet		last month	
Program.cs	Init projet		last month	
ProjetAnnuaire.csproj	Creation model + Fonction pour Affichage en tableau de la liste des s...		last month	
ProjetAnnuaire.csproj.user	update		last month	
ProjetAnnuaire.sln	update projet		last month	
Startup.cs	session + Acces admin avec mdp		10 days ago	
appsettings.Development.json	Init projet		last month	
appsettings.json	Init projet		last month	

Présentation du projet

Les principales fonctionnalités

Nous avons maintenant tous les outils à notre disposition pour commencer à coder notre projet.

Nous retrouvons dans le projet 3 parties :

- La partie visiteur qui est une liste des salariés avec la possibilité de voir le détail de chaque salarié et de trier cette liste selon des critères comme le nom, le site ou le service.
- La partie Admin qui est des CRUD (Create/Read/Update/Delete) sur les salariés, les sites et les services
- La partie accès Admin qui est un formulaire de connexion qui permet de gérer l'accès à la partie Admin.

Pour la partie accès Admin : nous avons juste une page qui se trouve dans Views/Login/Index.cshtml qui est un formulaire demandant un login et un mot de passe. A la validation du formulaire, il appelle la fonction SubmitLogin() dans le Contrôleur Login qui fait appel aux Services Login qui appelle le Model Admin et notamment la fonction GetAdminLogin() qui est une requête SQL qui permet de vérifier si le login et le mot de passe entrée correspond bien aux informations contenues en base de données. Si les informations sont correctes, le Service Login enregistre en session l'authentification.

Pour la partie visiteur : nous avons une page d'accueil et une page détail pour un salarié. La page d'accueil Views/Home/Index.cshtml comporte un tableau qui liste les salariés et 3 champs permettant de filtrer les informations du tableau avec la possibilité de voir le détail du salarié en cliquant sur un bouton qui renvoie vers la page détail. La page détail

Views/Home/ShowEmployee.cshtml est juste l'affichage du détail de tous les champs du salarié dans des balises <p>.

Pour l'affichage des salariés dans le tableau, le Contrôleur Home appelle la fonction GetEmployees() du model Employee qui grâce à une requête SQL liste tous les salariés, pour ce qui est des filtres, la même requête est appelé mais avec une condition pour chaque filtre ce qui change l'affichage.

"SELECT id, firstname, lastname, phone, cellphone, mail, idSite, idService FROM employee"

Annuaire Admin

Liste des Salariés

Nom	Prénom	Service	Site	
test	test	Comptabilité	Nantes	Voir le salarié
Vanhuyse	Nicolas	Comptabilité	Paris	Voir le salarié
aa	aa	Accueil	Toulouse	Voir le salarié

© 2021 - Annuaire

Page d'accueil du projet Annuaire

Pour l'affichage du détail d'un salarié, c'est le même procéder mais la fonction appelée est alors GetEmployee() et elle ne sort pas une liste mais juste les informations d'un seul salarié selon l'id entrée en condition dans la requête.

"SELECT id, firstname, lastname, phone, cellphone, mail, idSite, idService FROM employee where id = @id"

Annuaire Admin

Information sur le salarié :

Nom : test
Prénom : test
Téléphone fixe : 1212121212
Téléphone portable : 2222222222
Adresse mail : test@test.fr
Service : Comptabilité
Site : Nantes

© 2021 - Annuaire

Page d'un salarié du projet Annuaire

Pour la partie Admin : nous y avons accès seulement si nous retrouvons en session les informations correctes de l'authentification. Cette partie regroupe 3 CRUD (salarié, site, service). Le fonctionnement est le même pour les 3 parties.

Pour chaque CRUD nous avons l'affichage de la liste correspondante (salarié, site ou service) qui est la même méthode que sur la page d'accueil. Nous retrouvons à chaque fois un bouton qui permet la création, la modification ou la suppression.

Annuaire Admin

Gestion des Salariés

Ajouter un salarié

Nom	Prénom	Service	Site	
test	test	Comptabilité	Nantes	✎ ✖
Vanhuyse	Nicolas	Comptabilité	Paris	✎ ✖
aa	aa	Accueil	Toulouse	✎ ✖

© 2021 - Annuaire

Page de gestion (Salarié, site, service) du projet Annuaire

Pour la création, nous avons une page avec un formulaire qui reprend tous les champs de l'objet. A la validation du formulaire, la fonction Save() du Model correspondant est appelé par le Contrôleur et une requête INSERT SQL est alors exécutée ce qui créer l'objet en base.

"INSERT INTO employee (lastname, firstname, phone, cellphone, mail, idSite, idService) values (@lastname, @firstname, @phone, @cellphone, @mail, @idSite, @idService);"

Annuaire Admin

Ajout

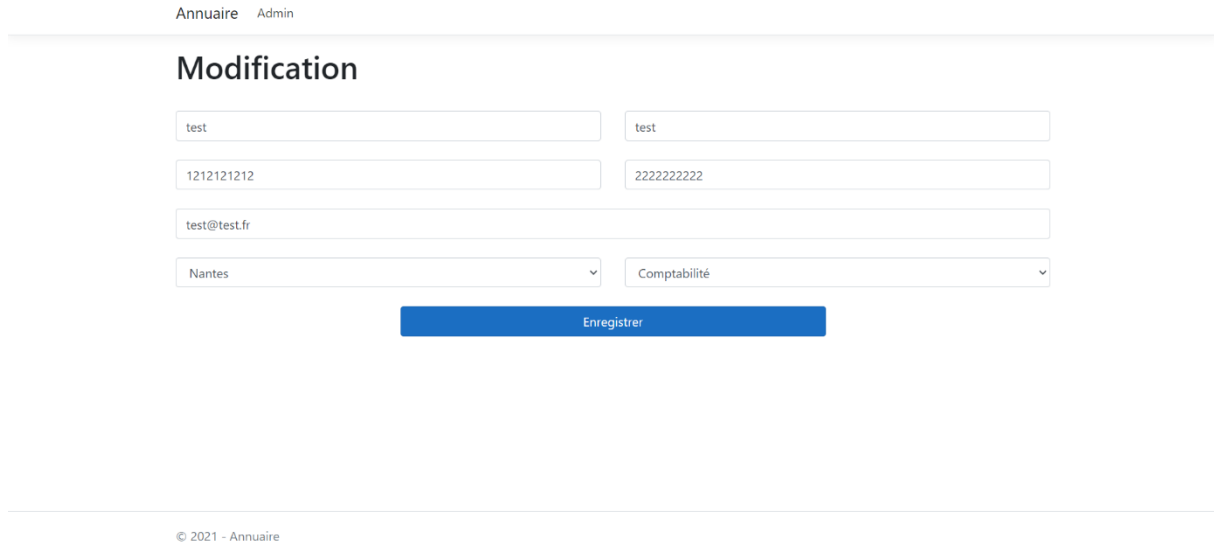
Enregistrer

© 2021 - Annuaire

Page d'ajout du projet Annuaire

Pour la modification, nous avons une page avec un formulaire qui reprend tous les champs de l'objet rempli grâce à la même fonction utilisée pour la page détail d'un salarié. A la validation du formulaire, le Contrôleur appelle la fonction Update() qui est une requête UPDATE SQL qui modifie les informations en base selon l'id rentrée en condition de la requête.

```
"Update employee set lastname=@lastname, firstname=@firstname, phone=@phone, cellphone=@cellphone, mail=@mail, idSite=@idSite, idService=@idService where id=@id"
```



Page de modification du projet Annuaire

Pour la suppression, un bouton sur chaque ligne dans le tableau permet au clique et à la fonction Delete() du model correspondant d'effectuer une requête DELETE SQL qui supprime de la base la ligne selon l'id rentrée en condition de la requête.

```
"DELETE FROM employee where id=@id"
```

Bilan

Difficultés rencontrées

L'environnement MVC n'est pas évident à prendre en main et cela m'a fortement ralenti dans la réalisation du projet.

La mise en place de Session est la partie où j'ai rencontré le plus de difficultés car j'ai entre autres oublié d'ajouter les services dans le Startup.cs ce qui me causé des erreurs où j'ai mis du temps à trouver d'où venait la cause.

Axes d'améliorations

La première chose qui pourrait être amélioré dans le projet est la sécurité. Je n'ai pas à l'heure d'aujourd'hui les connaissances nécessaires pour sécuriser un site web.

La partie Front du site est à revoir, l'utilisation de Bootstrap facilite la chose mais cela reste très basique et peu esthétique.

Conclusion

Le résultat final du projet correspond aux besoins définis dans le cahier des charges ainsi qu'aux objectifs annoncé au début du dossier. Le code est en objet, organisé correctement, commenté et versionné en utilisant l'outil Git.